REMARKS

Applicants respectfully request that the above-identified application be reexamined.

The January 12, 2007, Office Action ("Office Action") rejected Claims 21, 23, and 25-30 due to non-statutory subject matter, i.e., not describing a physical transformation outside of the computer. Claims 21, 23, and 25-30 have been amended to describe a physical transformation outside of the computer. More specifically, these claims have been amended to recite that the signature is stored in a log file. Thus, these claims will not be discussed further.

Claims 1, 5-9, 21, and 25-29 were rejected in the Office Action under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,751,794 ("McCaleb et al."). Claims 2-4 and Claims 22-24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over McCaleb et al. in view of Patent Application Publication No. US 2003/0090531 A1 ("Wong et al."). Claims 10 and 30 were rejected under 35 U.S.C. § 103(a) as being unpatentable over McCaleb et al. in view of Patent Application Publication No. 2004/0031030 ("Kidder et al.").

While applicants respectfully disagree with the rejections set forth in the Office Action, in order to advance the prosecution of this application, the claims remaining in this application have been amended to clarify one feature. Applicants respectfully submit that all of the claims remaining in this application (1-10 and 21-30) are clearly allowable in view of the teachings of the cited and applied references.

Prior to discussing in detail why applicants believe that all the claims in this application are allowable, a brief description of the disclosed subject matter and brief descriptions of the teachings of the cited and applied references are provided. The following discussions of the disclosed subject matter and the cited and applied references are not provided to define the scope or interpretation of any of the claims in this application. Instead, these discussions are provided to help the United States Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

Disclosed Subject Matter

A method for collecting information about the programs installed on a computer and the services provided by the computer and storing the information in log files is disclosed. In one exemplary implementation, the log file storage method collects three types of information. The first type is system information including, but not limited to, information about the computer operating system, hardware, and processor. The second type is information about the executables, i.e., executable programs used by the operating system, including, but not limited to, information about executables included in a defined set of folders stored on the computer and executables associated with services provided by the computer. The executables information includes attributes determined by the executables. The third type is application information about the application programs installed on the computer including linked executables. The application program information includes attributes determined by the application programs including the linked executables. The collected information is stored in a log file in a standard language. Further, a signature for each executable is also stored in the log file in a standard language. A signature for an executable is derived from a subset of the attributes associated with the executable.

In another exemplary implementation, the log file storage method of collecting and storing information about the applications installed on, and the services provided by, a computer for subsequent retrieval comprises enumerating executables associated with each application installed on, and each service provided by, the computer that has an associated executable. For each executable, information about the executable is extracted. The information includes a plurality of attributes regarding the executable. The extracted information is stored in a log file.

A signature for a combined set of attributes that include attributes from each of the executables is derived and stored in a log file.

The log files may be used to produce reports describing the applications, services, system information, and/or application compatibility information for the computing device. The signature allows log files containing certain attribute values, e.g., an application and its version to be more quickly identified.

U.S. Patent No. 6,751,794 (McCaleb et al.)

McCaleb et al. purportedly discloses a method for remotely updating software for a plurality of client systems. A client system sends a request for an upgrade to a server system. The request includes a unique identification that is recognized by the server system as belonging to the client system. In response, the server system sends an instruction to the client system that directs the client system to collect application information about the software application installed on the client system. The client system sends the application information to the server system. The server system performs a comparison between the application information about the software application and the most updated upgrade package for the software application. The server system sends the most updated upgrade package for the software application to the client system.

U.S. Patent Application Publication No. US 2003/0090531 (Wong et al.)

Wong et al. purportedly discloses a digital preservation system for accepting a digital data record as input. The digital data record is written in human-readable form onto a preservation quality medium. The digital preservation system preserves the data record in a human-readable form, along with an associated metadata record. Preserving the data record in a human-readable form allows the preserved data record to be readable independent of specific reading hardware. Hence, the preserved data record may be readable in the distant future.

U.S. Patent Application Publication No. 2004/0031030 (Kidder et al.)

Kidder et al. purportedly discloses a method and apparatus for facilitating hot upgrades of software components within a telecommunications network device through the use of signatures generated by a signature-generating program. After installation of a new software release within the network device, only those software components whose signatures do not match the signatures of corresponding and currently executing software components are upgraded. The signatures provide a way to accurately determine the upgrade status of each software component.

35 U.S.C. § 102(b) Rejection of Claims 1, 5-9, 21, and 25-30 based on McCaleb et al.

As amended, Claim 1 reads as follows:

> 1. A computer-implemented method of collecting and storing information about the programs installed on and the services provided by a computer for subsequent retrieval, comprising:
>
> (a) extracting from the computer information including, but not limited to, information about the computer operating system, hardware, and processor and storing the system information in a log file in a standardized language;
>
> (b) extracting from the computer executables information including, but not limited to, information about executables included in a defined set of folders stored on the computer and executables associated with services provided by the computer and storing the executables information in the log file, the executables information including attributes determined by the executables in a standardized language;
>
> (c) extracting from the computer information regarding the application programs installed on the computer including linked executables and storing the application program information in the log file in a standardized language, the application program information including attributes determined by the application programs including the linked executables; and
>
> (d) deriving a signature for each of the executables based on a subset of the attributes associated with the executable and storing the resultant signatures in the log file in a standardized language.

Remarks in the Office Action accompanying the rejection of Claim 1 appear to equate the client database of McCaleb et al. with the log file of the log file storage method recited in Claim 1. These remarks read as follows:

> ...extracting from the computer system information including, but not limited to, information about the computer operating system, hardware, and processor and storing the system information in a log file (column 4 lines 20-24, as a client database (log file) that includes information on software packages on the client system...).

Applicants respectfully disagree. A database, such as the client database of McCaleb et al., is not the same as a file that employs a standard language, such as XML. More specifically, while a database and a file are both storage facilities, i.e., both are capable of storing data that is searchable and retrievable, there are significant differences. A database is an application comprising both data and computer-executable instructions for managing the data. A file in a standard language comprises only data with no computer-executable instructions for managing the data. A database must be queried using a language programmed into the database to extract information from the database whereas a file is readable by any computer program capable of parsing the file format. An advantage of databases is the ability to provide complex cross referencing of the data in the databases. An advantage of files is that files are easily replicated and distributed. Another advantage of files is that files can be read by computer programs other than the computer program that stored the original data.

In the method of McCaleb et al., a server manages the installation of software patches for a plurality of computer programs running on a plurality of computers on a network. The server runs a database, gathers data about the software patch installations, stores the data in the database, retrieves data from the database, and uses the data retrieved from the database to track the software patch installations. In contrast, the claimed log file storage method gathers hardware and software information. The gathered information is stored in a file in a standard language, such as XML. Any computer program capable of parsing the standard language can retrieve and use the information. Hence, because the information is stored in a standard language file, the information can be retrieved and used by computers other than the computer that gathered and stored the data.

In contrast, McCaleb et al.'s data is not capable of being quickly retrieved and easily used by computers other than the server incorporating McCaleb et al.'s database and computers networked to that server because the data is stored in a database. A computer program that needs to retrieve the data in the database must query the database. Hence, a computer program that needs to use the data in the database must retrieve the data by interacting directly with the database or a copy of the database. Because databases are more difficult to replicate and distribute than files, the distribution of databases, and hence the data in the databases, is limited. Limited data distribution is not identified as a disadvantage in McCaleb et al. because the data is used by the McCaleb et al. server to manage the installation of software patches for a plurality of computer programs installed on each of the plurality of computers the server manages. That is, the data in the database used with only one closed system.

Limited distribution is a disadvantage in situations where multiple, and perhaps heterogeneous, networks must be managed. In such situations, the ease of distribution that standard language files provide is a distinct advantage. Those skilled in the art will understand that this advantage had been obscured because, traditionally, databases were used where complex cross referencing was required. The log file storage method provides both complex cross referencing and ease of distribution by using standard language, e.g., XML, files.

For the reasons discussed above, applicants submit that McCaleb et al. does not anticipate independent Claim 1 and hence Claim 1 is allowable. Further, Claims 5-9, which depend directly or indirectly from independent Claim 1, are submitted to be allowable for at least the reasons Claim 1 is allowable and for additional reasons derived from the additional recitations included in these claims.

Claim 21, as amended, reads as follows:

> A computer-implemented method of collecting and storing information about the applications installed on and the services provided by a computer for subsequent retrieval, comprising:
>
> enumerating executables associated with each application installed on and each service provided by the computer that has an associated executable;
>
> for each executable, extracting information about the executable, the information including a plurality of attributes regarding the executable and storing the information in a log file in a standardized language; and
>
> deriving a signature for a combined set of attributes including attributes from each of the executables and storing the signature in the log file in a standardized language.

Remarks in the Office Action regarding the rejection of Claim 21 state that:

> McCaleb anticipates a computer implemented method of collecting and storing information about the applications installed on and the services provided by a computer for subsequent retrieval (column 3, lines 60-65, client database tracks the installed software of the client system)....

As with amended Claim 1, amended Claim 21 recites a log file storage method that stores information in a log file in a standardized language, such as XML, and not, as in the McCaleb et al. method, in a client database. As a result, applicants submit that, like Claim 1, McCaleb et al. does not anticipate independent Claim 21 and, hence, Claim 21 is allowable. Further, Claims 25-29, which depend directly or indirectly from independent Claim 21, are submitted to be allowable for at least the reasons Claim 21 is allowable and for additional reasons derived from the additional recitations included in these claims.

Rejection of Claims 2-4 and 22-24 under 35 U.S.C. § 103(a) based on McCaleb et al. and Wong et al.

Remarks in the Office Action accompanying the rejection of Claims 2-4 and 22-24 state that McCaleb et al. does not explicitly teach storing information in an XML file. Applicants respectfully agree.

The remarks further state that Wong et al. teaches storing information in an XML file and conclude that therefore the claimed log file method would have been obvious to persons of ordinary skill in the art at the time the invention was made. Applicants respectfully disagree. Those skilled in the art will understand that until the log file storage method was developed, the advantage of providing both complex cross referencing and ease of distribution using standard language files, such as XML files, had been appreciated. Traditionally, complex cross-referencing databases were used. Applicants submit that it would not have been obvious to one skilled in the art at the time of invention combine the teachings of McCaleb et al and Wong et al. Thus, Claims 2-4 and 22-24 are submitted to be allowable in view of the teaching of McCaleb et al. and Wong et al.

Rejection of Claims 10 and 30 under 35 U.S.C. § 103(a) based on McCaleb et al. and Kidder et al.

Remarks in the Office Action accompanying the rejection of Claims 10 and 30 state that McCaleb et al. does not explicitly teach deriving a signature comprises generating a number from a subset utilizing a cyclic redundancy check. Applicants respectfully agree. The remarks further state that Kidder et al. teaches this limitation. Applicants respectfully disagree. While Kidder et al. recognizes that a cyclic redundancy check can be used to generate a signature, Kidder et al. teaches away from using a cyclic redundancy check to generate a signature. In particular, Kidder et al., paragraph 459, states that:

> A simple program such as a checksum or cyclic redundancy checking (CRC) program may be used to generate signature. The concern with such a simple program is that it may generate the same signature for a current software component and an upgrade of that component if the upgrade changes are not significant. Instead, a more robust program, such as a strong cryptographic program may be used to generate the signatures....

As a result, applicants submit it would not have been obvious to one skilled in the art at the time of invention to combine McCaleb et al and Kidder et al. Thus, Claims 10 and 30 are submitted to be allowable in view of the teaching of McCaleb et al. and Kidder et al.

CONCLUSION

In view of the foregoing amendments and remarks, applicants respectfully submit that all of the remaining claims in this application are allowable. Consequently, early and favorable action allowing these claims and passing this application to issue is respectfully solicited.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^PLLC

Gary S. Kindness
Registration No. 22,178
Direct Dial No. 206.695.1702

GSK/MFM:aew